

**PATENT APPLICATION  
DOCKET NO. 0100.9900210**

**In the United States Patent and Trademark Office**

**FILING OF A UNITED STATES PATENT APPLICATION**

**Title:**

**A VIDEO GRAPHICS MODULE CAPABLE OF BLENDING  
MULTIPLE IMAGE LAYERS**

**Inventors:**

**David I.J. Glen  
14 Glen Manor Drive  
Toronto, Ontario ME4 2X2  
Canada**

**Attorney of Record  
Timothy W. Markison  
Registration No. 33,534  
175 W. Jackson Blvd. – Suite 1015  
Chicago, Illinois 60604  
Phone (512) 894-3496  
Fax (512) 858-1105**

Express Mail Label No. EL454270483US

Date of Deposit: 1-13-00

I hereby certify that this paper is being deposited with the U.S. Postal Service "Express Mail Post Office to Addresses" service under 37 C.F.R. Section 1.10 on the 'Date of Deposit', indicated above, and is addressed to the Commissioner of Patents and Trademarks, Washington, D.C. 20231.

Name of Depositor: KRIS HERZOG  
(print or type)

Signature: Kris Herzog

5

## **A VIDEO GRAPHICS MODULE CAPABLE OF BLENDING MULTIPLE IMAGE LAYERS**

10

### **Technical Field of the Invention**

This invention relates generally to video graphic processing and more particularly to blending multiple image layers.

15

### **Background of the Invention**

20

25

Computers are known to include a central processing unit, system memory, interconnecting buses, audio processing circuitry, video graphics processing circuitry, and peripheral ports. The peripheral ports allow the central processing unit to communicate with external peripheral devices such as monitors, printers, external tape drives, etc. If the computer system includes basic video graphic processing, the video graphics processing circuitry receives graphics data from the central processing unit and prepares it for display on the monitor. Such graphics data is generated by the central processing unit while performing an application such as word processing, desktop operation, drawing application, presentation application, spreadsheet application, etc. In many computer systems, a cursor is provided that allows the user to utilize a graphical interface to select certain objects of an application.

30

Currently, the cursor may be generated in hardware as a monochrome cursor or as a color cursor using AND/XOR blending. Alternatively, the cursor may be generated in

software using any desired color depth or blending format. In particular, software generated color cursors using alpha blending are becoming more common and will be used more in future operating systems. For a hardware generated monochrome cursor, data is stored as 2 bits per pixel (bpp) in a frame buffer of the video graphics circuitry.

5 As is known, the frame buffer stores video graphics data that will subsequently be displayed. The 2 bpp are representative of four codes, where "00" indicates that the cursor is to be cursor color "0", which addresses a programmable register to retrieve the cursor color, which is normally set to black. A code of "01" indicates that the cursor is to be cursor color "1", which addresses a second programmable register to retrieve the color,  
 10 which is normally set to white. A code of "10" indicates that the cursor is to be transparent. A code of "11" indicates that the cursor is transparent with the background color inverted. As the cursor data is retrieved from the frame buffer, it is mixed, via a multiplexing function, with other video graphics data stored in the frame buffer to produce a composite image. Such a hardware implementation does not allow the  
 15 monochrome cursor to be alpha blended with other video graphics data.

Current hardware generated color cursors are AND/XOR blended with other video graphics data on a pixel by pixel basis. AND/XOR blending begins by ANDing each bit of video graphics data with an ANDing bit. The resultant of each ANDing  
 20 function is then exclusively ORed with a corresponding bit of the color cursor data to produce the blended data. Note that the color depth of the video graphics data and the color cursor data may be 8, 16, 24 bpp, etc. While this approach allows a color cursor to be blended with video graphics data, it does not allow alpha blending of the color cursor data with the video graphics data.

25 Software generation of a color cursor allows the cursor data to be alpha blended with underlying video graphics data to produce a composite image, which is stored in the frame buffer. As such, the software cursor is merged into the graphics layer. Such software alpha blending of a color cursor works well as long as the composite image is  
 30 not further blended with video data (e.g., a digitized television signal). When the

composite image is to be blended with video data, if the video layer is on top of the graphics layer, the cursor will be blocked. Further, if the cursor is blended with a key color (i.e., data that indicates placement of video in the display), the key color is altered, thus video data will not appear in the desired area of the display. Note that alpha

5   blending may be performed on a pixel-by-pixel basis using a pre-multiplied, or associated, alpha blend equation of  $P = (1-\alpha) * \text{graphics data (R,G,B)} + \text{cursor data (R, G, B)}$ , or an non-premultiplied alpha equation of  $P = (1-\alpha) * \text{graphics data} + \alpha * \text{cursor data}$ , where P equals the result pixel data, and where alpha is based on opacity.

10       Therefore, a need exists for a video graphics module that is backwards compatible with existing cursor implementations and is also capable of alpha blending multiple image layers wherein at least one of the image layers includes video data without the limitations of monochrome cursors, hardware color cursors using AND/XOR blending, and software color cursors.

### **Brief Description of the Drawings**

15       FIG. 1 illustrates a schematic block diagram of a video graphics module in accordance with the present invention;

20       FIG. 2 illustrates a schematic block diagram of another embodiment of a video graphics module in accordance with the present invention;

25       FIG. 3 illustrates a schematic block diagram of a mixing module of the video graphics module of FIG. 2;

      FIG. 4 illustrates a schematic block diagram of another mixing module of the video graphics module of FIG. 2; and

FIG. 5 illustrates a schematic block diagram of another embodiment of a video graphics module in accordance with the present invention.

### **Detailed Description of a Preferred Embodiment**

Generally, the present invention provides a video graphics module capable of blending multiple image layers. Such a video graphics module includes a plurality of video graphic pipelines, each of which is operable to process a corresponding image layer. One of the video graphic pipelines processes a foremost image layer. For example, the foremost image layer may be a hardware cursor. The video graphics module also includes a blending module that is operably coupled to the plurality of video graphic pipelines. The blending module blends, in accordance with a blending convention (e.g., AND/Exclusive OR blending and/or alpha blending), the corresponding image layers of each pipeline in a predetermined blending order to produce an output image. The blending module blends the foremost image layer such that it appears in a foremost position with respect to the other image layers. With such a video graphics module, a cursor may be alpha blended with graphics data and/or video data without loss of video data and without loss of the cursor.

The present invention can be more fully described with reference to FIGs. 1-5. FIG. 1 illustrates a schematic block diagram of a video graphics module 10 that includes a plurality of video graphic pipelines 12-18 and a blending module 20. The construct of the video graphics pipelines 12-18 is dependent on the type of data being represented by the corresponding image layers. For example, if image layer 22 is graphics data received from a central processing unit, the video graphics pipeline corresponds to a video graphics processor that receives the graphical information and produces the corresponding pixel information. Such a video graphics pipeline is found in ATI Technologies, Inc. All in Wonder Board and newer versions thereof. If the image layer is representative of video data, for example image layer 24, the corresponding video

graphics pipeline 16 may include a video decoder, video capture module, a YUV-to-RGB conversion module, and/or a scaler. The video decoder and video capture module are of a construct as found in the ATI Technologies, Inc. All in Wonder Board and versions thereof. In addition, the image layer may be a hardware cursor. For example, the  
 5 foremost image layer 26 may be a hardware cursor. As such, the video graphics pipeline 18 would be a hardware cursor pipeline as found in existing video graphic circuitry.

FIG. 1 further illustrates a graphical representation of the functionality of blending module 20. As shown, the video graphics pipeline 12 outputs text information (e.g.,  
 10 classic bike sale). The second video graphic pipeline 16 outputs an image of a bicycle. Such an image of the bicycle may be a still frame, or a video of a bicycle in motion. The third video graphics pipeline 18 outputs a tag having a sales price thereon. The blending module receives the corresponding pixel information for each of these images and blends them to produce an output image 28. The blending module 20 blends these images in  
 15 accordance with a blending convention, such as AND/Exclusive OR blending and/or an alpha blending equation. The blending module 20 also blends the image layers in a predetermined order such that the output image 28 has the foremost image layer 26 in a foreground position with respect to the other image layers. As such, the blending module 20 may blend the classic bike sale text with the image of the bicycle to have the classic  
 20 bike sale text in a foreground position with the bicycle as shown in output image 28-1, or have the bicycle in a foreground position with respect to the text, as shown in output image 28-2. Further, but not shown, the classic bike sale text and the image of the bicycle may be alpha blended together with the tag superimposed thereon.

25 If the blending module 20 is performing alpha blending, the blending module will use the premultiplied alpha blending equation or the non-premultiplied alpha blending equation. The alpha blending value may be a specific per pixel alpha value or a global alpha value and is done using one of a plurality of pixel depth. For example, the pixel depth may be 2, 8, 16, 24, or 32 bits per pixel. The AND/Exclusive OR blending may be  
 30 done with any of these pixel depths.

As one of average skill in the art will appreciate, the video graphics module 10 allows multiple image layers to be blended in a predetermined order using a blending convention to maintain a foremost image without loss of data of other image layers. This is particularly useful when a hardware cursor is to be alpha blended with other image layers. One of average skill in the art will further appreciate that the video graphics module 10 is backward compatible to process monochrome cursors, hardware or software color cursors.

FIG. 2 illustrates a schematic block diagram of another embodiment of the video graphics module 30. The video graphics module 30 includes a frame buffer 32, the plurality of video graphic pipelines 12-18, the blending module 20, and a digital-to-analog converter 40. The blending module 20 includes a first mixing module 34 and a second mixing module 38. As shown, the first video graphics pipeline 12 retrieves graphics data 42 from the frame buffer and provides processed graphics data 43 to the blending module 20. The second video graphics pipeline 16 retrieves video data 44 from the frame buffer 32 to provide processed video data 45 to the blending module 20. The third video graphics pipeline 18 retrieves hardware cursor data 46 from the frame buffer 32 and provides cursor data 47 to the blending module 20. Note that the graphics data, 42, the video data 44, and the hardware cursor 46 correspond to image layers that may have an RGB color base or a YUV color base. For an RGB color base, the video graphic pipelines 12-18 process the data in an RGB color base format. For a YUV color base, the video graphic pipelines 12-18 process the image layers in a YUV color base format.

The first mixing module 34 is operably coupled to receive the processed graphics data 43 from the video graphics pipeline 12 and the processed video data 45 from the video graphics pipeline 16. The first mixing module 34 mixes the signals to produce an intermediate blended image 48. In this configuration, the intermediate blended image 48 includes the processed video data 45 blended with the processed graphics data 43. As mentioned in the background section, the video data 44 is retrieved from a video capture

module based on color key information contained with the graphics data, which is often referred to as keying. Note that the keying process may be performed using the graphics data 42 and/or the video data 44. Further note that the keying may be performed at the end of pipelines 12 and/or 16 or at the beginning of the mixing module 34. Alternatively, the video data may be received directly from a video decoder based on the color key information without storage in the video capture module. By performing the mixing function on the processed video data 45 and the processed graphics data 43 prior to blending in the cursor data 47, the subsequent mixing of the hardware cursor will preserve the cursor data and the video data.

The second mixing module 38 receives the intermediate blended image 48 and the cursor data 47 from the third graphics pipeline 18. The second mixing module 38 mixes these images to produce a digital output 50. The mixing performed by the first and second mixing modules may be done utilizing the blending convention (i.e., the AND/Exclusive OR blending or alpha blending), wherein the data will be one of a plurality of pixel depths. The digital output 50 may be provided to an LCD panel, or other digital display module, or provided to the digital to analog converter 40. The digital to analog converter 40 converts the digital output 50 into an analog output 52 that may be provided to an analog display, such as a CRT.

FIG. 3 illustrates a schematic block diagram of the first mixing module 34. Note that the elements of the first mixing module 34 may be implemented as discrete processing modules and associated memories or via a single processing module and associated memory. The processing module may be a single processing device or a plurality of processing devices. Such a processing device may be a microprocessor, microcomputer, digital signal processor, central processing unit of a computer or work station, digital circuitry, state machine, and/or any device that manipulates signals (e.g., analog and/or digital) based on operational instructions. The memory may be a single memory device or a plurality of memory devices. Such a memory device may be a random access memory, read-only memory, floppy disk memory, hard drive memory,



extended memory, magnetic tape memory, zip drive memory and/or any device that stores digital information. Note that when the processing module implements one or more of its functions, via a state machine or logic circuitry, the memory storing the corresponding operational instructions is embedded within the circuitry comprising the state machine or logic circuitry.

The first mixing module 34 includes a first input 62, a second input 64, a blending module 60, a keyer 66, an alpha value calculation module 68, at least one general alpha value register 70 and a mode module 72. As configured, the first mixing module 34 may be in one of three alpha blending modes or an AND/XOR mixing mode. The first alpha blending mode involves a per pixel determination of the alpha blending data. The second alpha blending mode has involves alpha blending based on a global alpha blending value. The third alpha blending mode involves a keyed alpha blending value. The AND/XOR mixing mode has the blending module 60 mixing the color data using an AND/XOR function. Note that selection of any one of these modes may be done via a graphical user interface, a programmed decision of a particular application, or selected within features of the operating system.

For the AND/XOR mode, the first and second inputs 62 and 64, which may be buffers or terminal nodes, receive color data 43 and 45 from the respective pipelines and provide the color data to the blending module 60. The blending module 60, based on an input from the alpha value calculation module 68, performs an AND/XOR function upon the color data 43 and 45 to produce the intermediate blended image 48.

When the alpha blending mode 72 indicates using a global alpha blending value, the inputs 62 and 64 receive the processed graphics data 43 and the video data 45, which are provided to the blending module 60. The alpha value calculation module 68 retrieves at least one global alpha blending value from the general alpha register 70 and provides it to the blending module 60. The particular global alpha blending value retrieved will depend on the desired alpha blending affect and on which layers will be affected. For

example, a value of alpha A may be retrieved for both the graphics layer and the video layer, such that the same alpha blending is performed on both layers. As an additional example, a value of alpha B may be retrieved for the graphics layer and a value of alpha C may be retrieved for the video layer, such that the alpha blending affect on each layer is independent. The blending module 60 alpha blends, using the global alpha blending value, the graphics data 43 and the video data 45 to produce the intermediate blended image 48.

When the alpha blending mode 72 indicates using a per pixel alpha blending value, the inputs 62 and 64 receive the processed graphics data 43 and the video data 45, which are provided to the blending module 60. The first and second inputs 62 and 64 also receive the corresponding per pixel alpha blending value, which may be the same for the video layer and the graphics layer, or independent per pixel values. The per pixel alpha blending value(s) is provided to the alpha value calculation module 68, which provides it to the blending module 60. The blending module 60 alpha blends the graphics data 43 and the video data 45, on a per pixel basis using the per pixel alpha blending value, to produce the intermediate blended image 48. Note that the blending module 60 may be programmed to perform either a premultiplied alpha blending function or a non-premultiplied alpha blending function, which ever is correct for the current video or graphics input data.

When the alpha blending mode 72 indicates using a key alpha blending value, the inputs 62 and 64 receive the processed graphics data 43 and the video data 45, which are provided to the blending module 60. Optionally, the inputs 62 and 64 may also receive a corresponding per pixel alpha value. The processed graphics data 43, the processed video data 45, and the per pixel alpha value are provided to the keyer 66, which interprets the inputs to provide an alpha key indicator to the alpha calculation module 68. The alpha key indication, the alpha blending mode 72, and the global alpha value in register 70 are interpreted by the alpha calculation module 68 to determine a type of alpha blending. For example, alpha blend at 50%, fade in, fade out, etc. The alpha calculation module 68

retrieves the particular type of alpha blending from the mode module 72 and provides the corresponding alpha blending value(s) to the blending module 60. The blending module 60 alpha blends the graphics data 43 and the video data 45, on a per pixel basis using the corresponding per pixel alpha blending value(s), to produce the intermediate blended

5 image 48. Note that the graphics and video keys are mixed by the keyer 66 in a manner that is known in the art to produce a one-bit result per pixel. For example, 0 is for graphics and 1 for video. The alpha value calculation module 68 utilizes the one-bit result to “turn-on” a black circuit for one input layer (e.g., the video layer) and select a global alpha value for the other layer (e.g., the graphics layer). Further note that alpha

10 blending without blackening out the key color is not useful, as it would change the color of the selected layer. The mixing module 34 of Figure 3, however, allows fading in/out without changing the color.

The alpha value calculation module 68 may further produce a control signal that

15 causes one of the inputs of the blending module 60 to be received as a black signal. With one input blacked out, the other input may be phased in using a varying alpha value, as derived by the alpha value calculation module 68. Note that the varying alpha value is stored in the alpha register 70. Alternatively, the alpha value calculation module 68 may generate a black signal that is gated with one of the inputs to the blending module 60 to

20 achieve the blacking of one of the inputs. In this alternative, the alpha value calculation module 68 still generates the varying alpha value for the phase-in/phase-out effect.

The first mixing module 34 may further include a pair of multiplexors wherein the output of the multiplexors are operably coupled to the first and second inputs,

25 respectively. As such, each multiplexor is operably coupled to receive a plurality of image input layers and selects one of them for blending. The selection of one of the image input layers by each of the multiplexor is controlled by the alpha calculation module 68 to produce the desired blended image. Note that the multiplexors may be included in the first and second input modules 62 and 64.

FIG. 4 illustrates a schematic block diagram of the second mixing module 38. In this embodiment, the second mixing module 38 includes a first input 82, a second input module 84 and an cursor blending module 80. The first input module 82 is operably coupled to receive the intermediate blended image 48 and the second input module 84 is operably coupled to receive the cursor data 47. The first and second inputs are operably coupled to the cursor blending module 80 that blends the intermediate blended image 48 with the cursor data 47 to produce the digital output 50. The second input module 84 may include a pixel color depth conversion module to convert the pixel color depth of the hardware cursor data between M bits and N bits. For example, the pixel conversion may be between 2 bits to 8 bits, 8 to 16, 16 to 24, or to 32, or any combination thereof. Note that the pixel color depth conversion module would be utilized to match the hardware cursor pixel depth with the pixel depth of the intermediate blended image 48. Further note that the cursor blending module 80 may receive cursor mixing mode selection information which indicates the type of mixing to be performed. For example, the cursor mixing selection may be to use AND/XOR mixing, to use premultiplied alpha blending, or to use non-premultiplied alpha blending. Still further note that the cursor mixing mode select information may be received in conjunction with the cursor data 47.

FIG. 5 illustrates a schematic block diagram of another embodiment of the blending module 20. In this embodiment of the blending module 20, the resultant output images have an RGB color or a YUV color base. To achieve the multiple color base outputs, the blending module includes RGB to YUV conversion modules 90, 92 and 98. The blending module also includes an RGB blending module 94 and a YUV blending module 96. The RGB blending module 94 and YUV blending module 96 corresponds to the first mixing module 34 of FIG. 2 and FIG. 3 and includes similar functionality, but operate in the RGB color base for module 94 and the YUV color base for module 96.

As such, graphics data 42 received in an RGB color base is converted to a YUV color base via module 92 and video received in an YUV color base is converted to an RGB color base by module 90. The RGB color base graphics data 42 and the converted

video data 44 are mixed by the RGB blending module 94 to produce an intermediate blended image. The blending module 100 receives the intermediate blended image and the the hardware cursor data 46 to produce the RGB output image.

5           The video data 44 and the converted graphics data are mixed by the YUV blending module 96 to produce a YUV intermediate blended image. This blended image is mixed with a YUV converted representation of the hardware cursor data 46 to produce a YUV output image. The YUV output image is produced by blending module 102. As such, YUV inputs to YUV outputs do not need to be converted into RGB and back to  
10   YUV, which, if required to occur, would causes quality loses.

          The preceding discussion has presented multiple embodiments of a video graphics module that performs blending on multiple images without loss of underlying video data and preserving a foremost image. The video graphics module is particularly useful when  
15   alpha blending, or shadowing, a hardware cursor and the underlying video graphics data includes video data. As in previous implementations, if such were to be performed, the video data would be lost underneath the hardware cursor and shadow area. With the video graphics module of the present invention, the underlying video data is not lost, since it is blending with graphics data prior to blending with the cursor, thereby  
20   producing the desired alpha blending results. As one of average skill in the art will appreciate, other embodiments may be derived from the teachings of the present invention without deviating from the scope of the claims.